NETMOD Working Group
Internet-Draft
Updates: 8407 (if approved)
Intended status: Standards Track
Expires: 25 August 2022

Q. Wu
B. Claise
Huawei
P. Liu
Z. Du
China Mobile
M. Boucadair
Orange

21 February 2022

Self-Describing Data Object Tags in YANG Data Models
draft-ietf-netmod-node-tags-06

Abstract

This document defines a method to tag data objects that are associated with operation and management data in YANG modules. This method for tagging YANG data objects is meant to be used for classifying data objects from different YANG modules and identifying their characteristics data. Tags may be registered as well as assigned during the module definition, assigned by implementations, or dynamically defined and set by users.

This document also provides guidance to future YANG data model writers; as such, this document updates RFC 8407.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 August 2022.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Self-Describing Data Object Tags: Massive Data Object Collection Use Case 4
4. Data Object Tag Values 7
4.1. IETF Tags 7
4.2. Vendor Tags 7
4.3. User Tags 8
4.4. Reserved Tags 8
5. Data Object Tag Management 8
5.1. Module Design Tagging 8
5.2. Implementation Tagging 9
5.3. User Tagging 10
6. Data Object Tags Module Structure 10
6.1. Data Object Tags Module Tree 10
7. YANG Module 10
8. Guidelines to Model Writers 14
8.1. Define Standard Tags 14
9. IANA Considerations 15
9.1. YANG Data Object Tag Prefixes Registry 15
9.2. IETF YANG Data Object Tags Registry 15
9.3. Updates to the IETF XML Registry 17
9.4. Updates to the YANG Module Names Registry 18
10. Security Considerations 18
11. Acknowledgements 18
12. Contributors 19
13. References 19
13.1. Normative References 19
13.2. Informative References 20
Appendix A. NETCONF Example 21
Appendix B. Non-NMDA State Module 22
Appendix C. Targeted data object collection example 26

Appendix D. Changes between Revisions 29
Authors' Addresses 31

1. Introduction

The use of tags for classification and organization purposes is fairly ubiquitous, not only within IETF protocols, but globally in the Internet (e.g., "#hashtags"). For the specific case of YANG data models, a module tag is defined as a string that is associated with a module name at the module level [RFC8819].

Many data models have been specified by various Standards Developing Organizations (SDOs) and the Open Source community, and it is likely that many more will be specified. These models cover many of the networking protocols and techniques. However, the data objects defined by these technology-specific data models might represent a portion of fault, configuration, accounting, performance, and security (FCAPS) management information ([FCAPS]) at different levels and network locations, but also categorised in various different ways. Furthermore, there is no consistent classification criteria or representation for a specific service, feature, or data source.

This document defines self-describing data object tags and associates them with data objects within a YANG module, which:

* Provide dictionary meaning for specific targeted data objects.

* Indicate a relationship between data objects within the same YANG module or from different YANG modules.

* Identify key performance metric data objects and the absolute XPath expression identifying the element path to the node.

The self-describing data object tags can be used by the NETCONF [RFC6241] or RESTCONF [RFC8040] client to classify data objects from different YANG modules and identify characteristic data. In addition, these tags can provide input, instructions, or indications to selection filters and filter queries of configuration or operational state on a server based on these data object tags (e.g.,

return specific objects containing operational state related to system-management). NETCONF clients can discover data objects with self-describing data object tags supported by a NETCONF server by means of the <get-schema> operation (Section 3.1 of [RFC6022]). The self-describing data object tag capability can also be advertised using the capability notification model [RFC9196] by the NETCONF server or some websites where offline documents are kept. Similar to YANG module tags defined in [RFC8819], these data object tags may be registered or assigned during the module definition, assigned by implementations, or dynamically defined and set by users.

This document defines a YANG module [RFC7950] that augments the module tag model [RFC8819] and provides a list of data object entries to add or remove self-describing tags as well as to view the set of self-describing tags associated with specific data objects within YANG modules.

This document defines three extension statements to indicate self-describing tags that should be added by the module implementation automatically (i.e., outside of configuration).

This document also defines an IANA registry for tag prefixes and a set of globally assigned tags (Section 9).

Section 8 provides guidelines for authors of YANG data models. This document updates [RFC8407].

The YANG data model in this document conforms to the Network Management Datastore Architecture defined in [RFC8342].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

The meanings of the symbols in tree diagrams are defined in [RFC8340].

3. Self-Describing Data Object Tags: Massive Data Object Collection Use Case

Among data object tags, the 'opm' (object, property, metric) tags can be used to tackle massive data object collections, indicate relationships between data objects, and capture YANG data objects associated with YANG-modelled performance metrics data. An example is depicted in Figure 1.

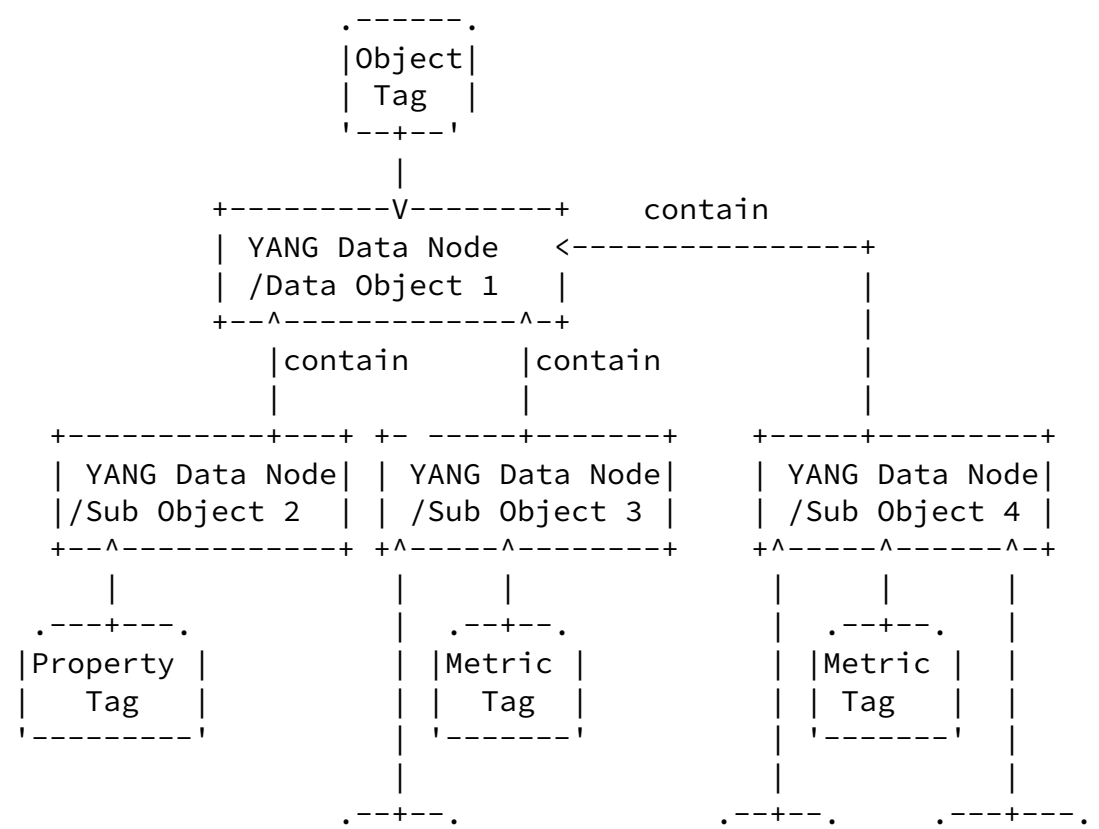


Figure 1: The Relation between Object, Property, and Metric

In Figure 1, data objects can contain other data objects called 'subobjects'. Both object and subobjects can be modeled as YANG data nodes [RFC7950].

Data objects that contain other data objects can be one of type 'container', 'leaf-list', or 'list' and are tagged with the object tag.

A subobject tagged with the property tag is a 'leaf' node.

Subobjects tagged with the metric tag can be one of 'container', 'leaf-list', 'list', or 'leaf' data node.

A data object may contain one single object tag, or one single property tag, or one single metric tag. The data object tagged with the metric tag also can have one or multiple Metric Type tags and/or one single multi-source tag.

The use of 'opm' tags is meant to help filter discrete categories of YANG data objects scattered across the same or different YANG modules that are supported by a device and capture all network performance data or all property data in a single view of the data. In the example shown in Figure 2, the 'tunnel-svc' data object is a container node defined in a 'tunnel-pm' module and can be seen as the root object for property tagged subobjects (e.g., 'tunnel-svc'/'create-time') and metric tagged subobjects (e.g., 'tunnel-svc'/'avg-latency'). The 'name', 'create-time', and 'modified-time' are property tagged subobjects under 'tunnel-svc' container. The 'avg-latency' and 'packet-loss' metrics are tagged subobjects under 'tunnel-svc' container node. Consider the 'tunnel-svc' data object and the 'tunnel-svc/name' data object as an example: the 'tunnel-svc' data object has one single object tag (i.e., 'ietf:object'), while the 'tunnel-svc/name' data object has one single property subobject

tag (i.e., 'ietf:property'). In addition, not all metric subobjects need to be tagged (e.g., define specific categories, such as loss-related metric subobjects need to be tagged with a metric-type tag which can further reduce amount data to be fetched).

Data Object	Object Tag	Property Tag	Metric Tag	Multi-Source Tag
tunnel-svc	ietf: object			
tunnel-svc/name		ietf: property		
tunnel-svc/create-time		ietf: property		
tunnel-svc/modified-time		ietf: property		
tunnel-svc/avg-latency			ietf: metric	non-agg
tunnel-svc/packet-loss			ietf: metric	non-agg
tunnel-svc/min-latency			ietf: metric	non-agg
tunnel-svc/max-latency			ietf: metric	non-agg

Figure 2: Example of OPM Tags Used in the YANG Module

If data objects in YANG modules are adequately tagged and learnt by the client from a server, the client can retrieve paths to all targeted data objects and then use an XPath query defined in [RFC8639][RFC8641] to list all tagged data objects which reflect the network characteristics.

4. Data Object Tag Values

All data object tags SHOULD begin with a prefix indicating who owns their definition. An IANA registry (Section 9.1) is used to register data object tag prefixes. Initially, three prefixes are defined.

No further structure is imposed by this document on the value following the registered prefix, and the value can contain any YANG type 'string' characters except carriage returns, newlines, tabs, and spaces.

Except for the conflict-avoiding prefix, this document is purposefully not specifying any structure on (i.e., restricting) the tag values. The intent is to avoid arbitrarily restricting the values that designers, implementers, and users can use. As a result of this choice, designers, implementers, and users are free to add or not add any structure they may require to their own tag values.

4.1. IETF Tags

An IETF tag is a data object tag that has the prefix "ietf:".

All IETF data object tags are registered with IANA in the registry defined in Section 9.2.

4.2. Vendor Tags

A vendor tag is a tag that has the prefix "vendor:".

These tags are defined by the vendor that implements the module, and are not registered with IANA. However, it is RECOMMENDED that the vendor includes extra identification in the tag to avoid collisions, such as using the enterprise or organization name following the "vendor:" prefix (e.g., vendor:example.com:vendor-defined-classifier).

4.3. User Tags

A user tag is any tag that has the prefix "user:". For the avoidance of confusion, the colon (":") when it appears for the first time, is always assumed to be the separator between a prefix and the rest of the tag. And so, when a user tag does not have a prefix, it MUST NOT contain a colon.

These tags are defined by a user/administrator and are not meant to be registered with IANA. Users are not required to use the "user:" prefix; however, doing so is RECOMMENDED as it helps avoid collisions.

4.4. Reserved Tags

Any tag not starting with the prefix "ietf:", "vendor:", or "user:" is reserved for future use (Section 9.1).

These tag values are not invalid, but simply reserved in the context of specifications (e.g., RFCs).

5. Data Object Tag Management

Tags may be associated with a data object within a YANG module in a number of ways. Typically, tags may be defined and associated at the module design time, at implementation time without the need of a live server, or via user administrative control. As the main consumers of data object tags are users, users may also remove any tag from a live server, no matter how the tag became associated with a data object within a YANG module.

5.1. Module Design Tagging

A data object definition MAY indicate a set of data object tags to be added by a module's implementer. These design time tags are indicated using a set of extension statements which include:

opm-tag extension statement: Classifies management and operation data into object, property subobject, and metric subobject categories.

Both objects and subobjects can be modeled as YANG data nodes [RFC7950]. Data objects that contain other data objects can be one of type 'container', 'leaf-list', and 'list' and are tagged with object tag. A subobject tagged with the property tag is a 'leaf' node. Subobjects tagged with the metric tag can be one of 'container', 'leaf-list', 'list', or 'leaf' data node. A data object contains one single object tag, one single property tag, or one single metric tag.

A data object tagged with metric tag also can have one or multiple Metric type tag and/or one single multi-source tag. See the examples depicted in Figure 2 and Figure 4.

metric-type extension statement: Provides metric data objects classifications (e.g., loss, jitter, delay, counter, gauge, summary, unknown) within the YANG module.

multi-source-tag extension statement: Identifies multi-source aggregation type (e.g., aggregated, non-aggregated) related to a metric subobject.

The 'aggregated' multi-source aggregation type allows a large number of measurements on metric subobjects from different sources of the same type (e.g., line card, each subinterface of aggregated Ethernet interface) to be combined into aggregated statistics and report as one metric subobject.

The 'non-aggregated' multi-source aggregation type allows measurement from each source of the same type (e.g., line card, each subinterface of aggregated Ethernet interface) to be reported separately.

Among these extension statements, the 'metric-type' and 'multi-source' tag extension statements are context information that can be used to correlate data objects from the different modules.

If the data node is defined in an IETF Standards Track document, the data object tags MUST be IETF Tags (Section 4.1). Thus, new data objects can drive the addition of new IETF tags to the IANA registry defined in Section 9.2, and the IANA registry can serve as a check against duplication.

5.2. Implementation Tagging

An implementation MAY include additional tags associated with data objects within a YANG module. These tags SHOULD be IETF

(Section 4.1) or vendor tags (Section 4.2).

5.3. User Tagging

Data object tags of any kind, with or without a prefix, can be assigned and removed by the user from a server using normal configuration mechanisms. In order to remove a data object tag from the operational datastore, the user adds a matching "masked-tag" entry for a given data object within the 'ietf-data-object-tags' module.

6. Data Object Tags Module Structure

6.1. Data Object Tags Module Tree

The tree associated with the "ietf-data-object-tags" module is as follows:

```
module: ietf-data-object-tags
augment /tags:module-tags/tags:module:
  +--rw data-object-tags
     +--rw data-object* [name]
        +--rw name  nacm:node-instance-identifier
        +--rw tag*         tags:tag
        +--rw masked-tag*  tags:tag
```

Figure 3: YANG Module Tags Tree Diagram

7. YANG Module

This module imports types from [RFC8819],[RFC8341].

```
<CODE BEGINS> file "ietf-data-object-tags@2022-02-04.yang"
module ietf-data-object-tags {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-data-object-tags";
  prefix ntags;

  import ietf-netconf-acm {
    prefix nacm;
    reference
```

```
    "RFC 8341: Network Configuration Access Control
               Model";
  }
  import ietf-module-tags {
    prefix tags;
    reference
    "RFC 8819: YANG Module Tags ";
  }
```



```
  organization
    "IETF NetMod Working Group (NetMod)";
  contact
   "WG Web:  <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Editor: Qin Wu
            <mailto:bill.wu@huawei.com>

    Editor: Benoit Claise
            <mailto:benoit.claise@huawei.com>

    Editor: Peng Liu
            <mailto:liupengyjy@chinamobile.com>

    Editor: Zongpeng Du
            <mailto:duzongpeng@chinamobile.com>

    Editor: Mohamed Boucadair
            <mailto:mohamed.boucadair@orange.com>";
  description
    "This module describes a mechanism associating self-describing
     tags with YANG data object within YANG modules. Tags may be IANA
     assigned or privately defined.

     Copyright (c) 2022 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
```

```
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX
      (https://datatracker.ietf.org/html/rfcXXXX); see the RFC itself
      for full legal notices.";

    revision 2022-02-04 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: Self-Describing Data Object Tags in YANG Data
                   Models";
    }

    extension opm-tag {
```



```
      argument tag;
      description
        "The argument 'tag' is of type 'tag'. This extension statement
         is used by module authors to indicate the opm tags that should
         be added automatically by the system. 'opm-tag' is used to
         classify operation and management data objects into the three
         categories, object, property, and metric. Data Object can
         contain other data objects called subobjects. Both object and
         subobjects can be modeled as data nodes. A data object
         tagged with object tag can be one of container, leaf-list, or
         list. A data object tagged is with the property tag is a leaf
         node. The data object tagged with the metric tag can be one of
         container, leaf-list, list, or leaf. A data objects tagged
         with either property tag or metric tag are subobjects
         belonging to a specific root data object. Each data object may
         contain one single object tag, or one single property tag,
         or one   single metric tag (these tags are mutually
         exclusive). As such,  the origin of the value for the
         pre-defined tags should be set to 'system'.";
    }

    extension metric-type {
      argument tag;
      description
        "The argument 'tag' is of type 'tag'. The metric type can be
```

```
        used to provide metric data object classification
        (e.g., loss, jitter, packet loss, counter, gauge,
        summary, unknown) within a YANG module.";
    }

    extension multi-source-tag {
      argument tag;
      description
        "The argument 'tag' is of type 'tag'. The multi-source-tag can
        be used to identify multi-source aggregation type
        (e.g., aggregated,  non-aggregated) related to a metric
        subobject.

        The 'aggregated' multi-source aggregation type allows a large
        number of measurements on metric subobjects from different
        sources of the same type (e.g., line card, each subinterface of
        an aggregated Ethernet interface) to be combined into
        aggregated statistics and reported as one metric subobject
        value.

        The 'non-aggregated'  multi-source aggregation type allows
        measurement from each source of the same type (e.g., line
        card, each subinterface of an aggregated Ethernet interface) to
```



```
        be reported separately.";
     }

    augment "/tags:module-tags/tags:module" {
      description
        "Augment the Module Tags module with data object tag
         attributes.";
      container data-object-tags {
        description
          "Contains the list of data objects and their associated data
           object tags.";
        list data-object {
          key "name";
          description
            "Includes a list of data objects and their associated data
             object tags.";
          leaf name {
            type nacm:node-instance-identifier;
```

```
            mandatory true;
            description
              "The YANG data object name.";
          }
          leaf-list tag {
            type tags:tag;
            description
              "Lists the tags associated with the data object within
               the YANG  module.

               See the IANA 'YANG Data Object Tag Prefixes' registry
               for reserved prefixes and the IANA 'IETF YANG Data
               Object Tags' registry for IETF tags.

               The 'operational' state view of this list is
               constructed using the following steps:

               1) System tags (i.e., tags of 'system' origin) are
                  added.
               2) User configured tags (i.e., tags of 'intended'
                  origin) are added.
               3) Any tag that is equal to a masked-tag is removed.";
            reference
              "RFC XXXX: Self-Describing Data Object Tags in YANG Data
                         Models, Section 9";
          }
          leaf-list masked-tag {
            type tags:tag;
            description
              "The list of tags that should not be associated with the
```



```
               data object within the YANG module. The user can remove
               (mask) tags from the operational state datastore by
               adding them to this list. It is not an error to add tags
               to this list that are not associated with the data
               object within YANG  module, but they have no operational
               effect.";
          }
        }
      }
    }
  }
```

<CODE ENDS>

8. Guidelines to Model Writers

This section updates [RFC8407].

8.1. Define Standard Tags

A module MAY indicate, using data object tag extension statements, a set of data object tags that are to be automatically associated with data object within the module (i.e., not added through configuration).

```
module example-module-A {
  //...
  import ietf-data-node-tags { prefix ntags; }

  container top {
    ntags:opm-tag "ietf:object";
    list X {
      leaf foo {
         ntags:opm-tag "ietf:property";
      }
      leaf bar {
        ntags:opm-tag "ietf:metric";
      }
    }
  }
  // ...
}
```

Figure 4: An Example of Data Object Tag

The module writer can use existing standard data object tags, or use new data object tags defined in the data object definition, as appropriate. For IETF standardized modules, new data object tags MUST be assigned in the IANA registry defined in Section Section 9.2.

9. IANA Considerations

9.1. YANG Data Object Tag Prefixes Registry

This document requests IANA to create "YANG Data Object Tag Prefixes" subregistry in "YANG Data Object Tag" registry.

This registry allocates tag prefixes. All YANG Data Object Tags should begin with one of the prefixes in this registry.

Prefix entries in this registry should be short strings consisting of lowercase ASCII alpha-numeric characters and a final ":" character.

The allocation policy for this registry is Specification Required [RFC8126]. The Reference and Assignee values should be sufficient to identify and contact the organization that has been allocated the prefix. There is no specific guidance for the Designated Expert and there is a presumption that a code point should be granted unless there is a compelling reason to the contrary.

The initial values for this registry are as follows:

Prefix	Description	Reference	Assignee
ietf:	IETF Tags allocated in the IANA IETF YANG Data Object Tags registry	[This document]	IETF
vendor:	Non-registered tags allocated by the module's implementer.	[This document]	IETF
user:	Non-registered tags allocated by and for the user.	[This document]	IETF

Figure 5: Table 1

Other standards organizations (SDOs) wishing to allocate their own set of tags should request the allocation of a prefix from this registry.

9.2. IETF YANG Data Object Tags Registry

This document requests IANA to create "IETF OPM Tags","IETF Metric Type Tags","IETF Multiple Source Tags" three subregistries in "YANG Data Object Tag" registry. These 3 subregistries appear below "YANG Data Object Tag Prefixes" registry.

Three subregistries allocate tags that have the registered prefix "ietf:". New values should be well considered and not achievable through a combination of already existing IETF tags.

The allocation policy for these three subregistries is IETF Review [RFC8126]. The Designated Expert is expected to verify that IANA assigned tags conform to Net-Unicode as defined in [RFC5198], and shall not need normalization.

The initial values for these three subregistries are as follows:

OPM Tag	Description	Reference
ietf:object	Represents Root object containing other data objects (e.g., interfaces)	[This document]
ietf:property	Represents a property data object(e.g., ifindex) associated with a specific root object (e.g., interfaces)	[This document]
ietf:metric	Represent metric data object(e.g., ifstatistics) associated with specific root object(e.g., interfaces)	[This document]

Metric Type Tag	Description	Reference
ietf:delay	Represents the delay metric group to which the metric data objects belong to.	[This document]
ietf:jitter	Represents the jitter metric group to which the metric data objects belong to.	[This document]
ietf:loss	Represents the loss metric group to which the metric data objects belong to.	[This document]
ietf:counter	Represents any metric value associated with a metric	

	data object that monotonically increases over time, starting from zero.	[This document]
ietf:gauge	Represents current measurements associated with a metric data object that may increase, decrease or stay constant.	[This document]
ietf:summary	Represents the metric value associated with a metric data object that measures distributions of discrete events without knowing predefined range.	[This document
ietf:unknown	Represents the metric value associated with metric data object that can not determine the type of metric.	[This document

Multiple Source Tag	Description	Reference
ietf:agg	Relates to multiple sources aggregation type (i.e., aggregated statistics)	[This document]
ietf:non-agg	Relates to multiple sources aggregation type (i.e., non-aggregated statistics)	[This document]

Figure 6: Table 2

9.3. Updates to the IETF XML Registry

This document registers the following namespace URI in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-data-object-tags
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

9.4. Updates to the YANG Module Names Registry

This document registers the following YANG module in the YANG Module Names registry [RFC6020] within the "YANG Parameters" registry:

name: ietf-data-object-tags
namespace: urn:ietf:params:xml:ns:yang:ietf-data-object-tags
prefix: ntags
reference: RFC XXXX
maintained by IANA: N

10. Security Considerations

The YANG module specified in this document defines schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content, e.g., the presence of tags may reveal information about the way in which data objects are used and therefore providing access to private information or revealing an attack vector should be restricted. Note that appropriate privilege and security levels need to be applied to the addition and removal of user tags to ensure that a user receives the correct data.

This document adds the ability to associate data object tag meta-data with data object within the YANG modules. This document does not

define any actions based on these associations, and none are yet defined, and therefore it does not by itself introduce any new security considerations.

Users of the data object tag meta-data may define various actions to be taken based on the data object tag meta-data. These actions and their definitions are outside the scope of this document. Users will need to consider the security implications of any actions they choose to define, including the potential for a tag to get 'masked' by another user.

11. Acknowledgements

The authors would like to thank Ran Tao for his major contributions to the initial modeling and use cases.

The authors would also like to acknowledge the comments and suggestions received from Juergen Schoenwaelder, Andy Bierman, Lou Berger,Jaehoon Paul Jeong, Wei Wang, Yuan Zhang, Ander Liu, YingZhen Qu, Boyuan Yan, Adrian Farrel, and Mahesh Jethanandani.

12. Contributors

Liang Geng
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: gengliang@chinamobile.com

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <https://www.rfc-editor.org/info/rfc8407>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8819] Hopps, C., Berger, L., and D. Bogdanovic, "YANG Module Tags", RFC 8819, DOI 10.17487/RFC8819, January 2021, <https://www.rfc-editor.org/info/rfc8819>.

13.2. Informative References

[FCAPS] International Telecommunication Union, "X.700 : Management framework for Open Systems Interconnection (OSI) for CCITT applications", , September 1992, <http://www.itu.int/rec/T-REC-X.700-199209-I/en>.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[RFC6022] Scott, M. and M. Bjorklund, "YANG Module for NETCONF Monitoring", RFC 6022, DOI 10.17487/RFC6022, October 2010, <https://www.rfc-editor.org/info/rfc6022>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641,

September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC9195] Lengyel, B. and B. Claise, "A File Format for YANG Instance Data", RFC 9195, DOI 10.17487/RFC9195, February 2022, <https://www.rfc-editor.org/info/rfc9195>.

[RFC9196] Lengyel, B., Clemm, A., and B. Claise, "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications", RFC 9196, DOI 10.17487/RFC9196, February 2022, <https://www.rfc-editor.org/info/rfc9196>.

Appendix A. NETCONF Example

The following is a NETCONF example result from a query of the data object tags list. For the sake of brevity only a few module and associated data object results are provided. The example uses the folding defined in [RFC8792].

```
================ NOTE: '\' line wrapping per RFC 8792 ================

<ns0:data xmlns:ns0="urn:ietf:params:xml:ns:netconf:base:1.0">
 <t:module-tags xmlns:t="urn:ietf:params:xml:ns:yang:ietf-module-tags">
    <t:module>
       <t:name>ietf-interfaces</t:name>
       <s:data-object-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf-\
data-object-tags">
        <s:data-object>
         <s:name>/if:interfaces/if:interface</s:name>
         <s:tag>ietf:object</s:tag>
        </s:data-object>
       <s:data-object>
```



```
         <s:name>/if:interfaces/if:interface/if:last-change</\
s:name>
         <s:tag>ietf:property</s:tag>
        </s:data-object>
         <s:data-object>
```

```
          <s:name>
           /if:interfaces/if:interface/if:statistics/if:in-errors
          </s:name>
          <s:tag>ietf:metric</s:tag>
          <s:tag>ietf:loss</s:tag>
          <s:tag>ietf:non-agg</s:tag>
         </s:data-object>
       </s:data-object-tags>
     </t:module>
     <t:module>
        <t:name>ietf-ip</t:name>
        <s:data-object-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf\
 -data-object-tags">
        <s:data-object>
          <s:name>/if:interfaces/if:interface/ip:ipv4</s:name>
          <s:tag>ietf:object</s:tag>
        </s:data-object>
        <s:data-object>
          <s:name>/if:interfaces/if:interface/ip:ipv4/ip:enable\
 </s:name>
          <s:tag>ietf:property</s:tag>
        </s:data-object>
         <s:data-object>
         <s:name>/if:interfaces/if:interface/ip:ipv4/ip:mtu</s:name>
         <s:tag>ietf:metric</s:tag>
         <s:tag>ietf:non-agg</s:tag>
        </s:data-object>
       </s:data-object-tags>
     </t:module>
   </t:module-tags>
 </ns0:data>
```

Figure 7: Example NETCONF Query Output

Appendix B. Non-NMDA State Module

As per [RFC8407], the following is a non-NMDA module to support viewing the operational state for non-NMDA compliant servers.

```
<CODE BEGINS> file "ietf-data-object-tags-state@2022-02-03.yang"
module ietf-data-object-tags-state {
  yang-version 1.1;
  namespace
        "urn:ietf:params:xml:ns:yang:ietf-data-object-tags-state";
  prefix ntags-s;

  import ietf-netconf-acm {
    prefix nacm;
    reference
    "RFC 8341: Network Configuration Access Control
               Model";
  }
  import ietf-module-tags {
    prefix tags;
    reference
    "RFC 8819: YANG Module Tags ";
  }
  organization
    "IETF NetMod Working Group (NetMod)";

  contact
   "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List:<mailto:netmod@ietf.org>

    Editor: Qin Wu
            <mailto:bill.wu@huawei.com>

    Editor: Benoit Claise
            <mailto:benoit.claise@huawei.com>

    Editor: Peng Liu
            <mailto:liupengyjy@chinamobile.com>

    Editor: Zongpeng Du
            <mailto:duzongpeng@chinamobile.com>

    Editor: Mohamed Boucadair
            <mailto:mohamed.boucadair@orange.com>";

  description
    "This module describes a mechanism associating self-describing
     tags with YANG data object within YANG modules. Tags may be
     IANA assigned or privately defined.

     Copyright (c) 2022 IETF Trust and the persons identified as
     authors of the code. All rights reserved.
```

```
        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX
        (https://datatracker.ietf.org/html/rfcXXXX); see the RFC
        itself for full legal notices.";

    revision 2022-02-04 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: Self-Describing Data Object Tags in YANG Data
                   Models";
    }

    extension opm-tag {
      argument tag;
      description
        "The argument 'tag' is of type 'tag'. This extension
         statement is used by module authors to indicate the opm tags
         that should be added automatically by the system. 'opm-tag'
         is used to classify operation and management data objects
         into the three categories, object, property, and metric.
         Data Object can contain other data objects called subobjects.
         Both object and subobjects can be modeled as data nodes. The
         Data Object tagged with object tag can be one of container,
         leaf-list and list. The Data Object tagged with the Property
         tag is a leaf node. The Data Object tagged with the Metric
         tag can be one of type container, leaf-list, list, leaf. The
         Data objects tagged with either property tag or metric tag
         are subobjects belonging to a specific root data object. Each
         Data Object may contain one single object tag, or one single
         property tag, or one single metric tag (these tags are
         mutually exclusive). As such, the origin of the value for the
         pre-defined tags should be set to 'system'.";
    }
    extension metric-type {
      argument tag;
```

```
    description
      "The argument 'tag' is of type 'tag'.The metric-type can be
       used to provide metric subobject classification
       (e.g., loss, jitter, packet loss, guage, counter, histogram,
       unknow, etc.) within the YANG module.";
  }
```



```
  extension multi-source-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'. The multi-source tag can
       be used to identify multi-source aggregation type (e.g.,
       aggregated, non-aggregated) related to a metric subobject.

      The 'aggregated' multi-source aggregation type allows a large
      number of measurements on metric subobjects from different
      sources of the same type (e.g., line card, each subinterface
      of aggregated Ethernet interface) to be combined into
      aggregated statistics and reported as one metric subobject
      value.

      The 'non-aggregated'multi-source aggregation type
      allows measurement from each source of the same type
      (e.g., line card, each subinterface of aggregated Ethernet
      interface) to be reported separately.";
   }

  augment "/tags:module-tags/tags:module" {
    description
      "Augments the Module Tags module with data object tag
       attributes.";
    container data-object-tags {
      config false;
      status deprecated;
      description
        "Contains the list of data objects and their
         associated self describing tags.";
      list data-object {
        key "name";
        status deprecated;
        description
          "Lists the data objects and their associated self
```

```
   describing tags.";
 leaf name {
   type nacm:node-instance-identifier;
   mandatory true;
   status deprecated;
   description
     "The YANG data object name.";
 }
 leaf-list tag {
   type tags:tag;
   status deprecated;
   description
     "Tags associated with the data object within the
```

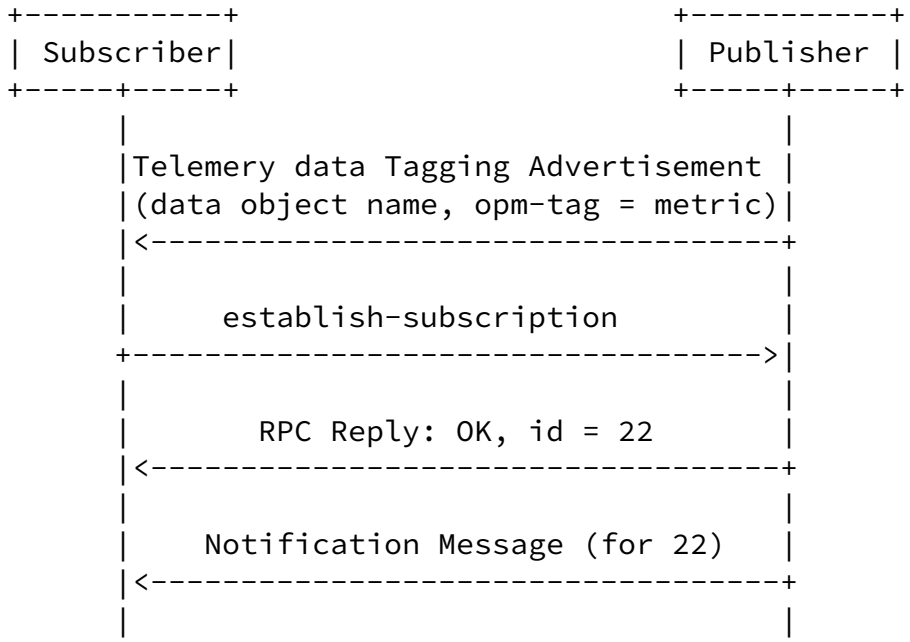


```
      YANG module. See the IANA 'YANG Data Object Tag
      Prefixes' registry for reserved prefixes and the
      IANA 'IETF YANG Data Object Tags'registry for
      IETF tags.

      The 'operational' state view of this list is
      constructed using the following steps:

      1) System tags (i.e., tags of 'system' origin) are
         added.
      2) User configured tags (i.e., tags of 'intended'
         origin) are added.
      3) Any tag that is equal to a masked-tag is removed.";
   reference
     "RFC XXXX: Self-Describing Data Object Tags in YANG Data
                Models, Section 9";
 }
 leaf-list masked-tag {
   type tags:tag;
   status deprecated;
   description
     "The list of tags that should not be associated with the
      data object within the YANG module. The user can remove
      (mask) tags from the operational state datastore by
      adding them to this list. It is not an error to add
      tags to this list that are not associated with the
      data object within YANG module, but they have no
      operational effect.";
```

```
          }
        }
      }
    }
  }
<CODE ENDS>
```

Appendix C. Targeted data object collection example

The following provides targeted data object collection example which helps reduce amount of data to be fetched. The subscription "id" values of 22 used below is just an example. In production, the actual values of "id" might not be small integers.



The publisher advertises telemetry data object capability to the subscriber to instruct the receiver to subscribe tagged data object (e.g., performance metric data object) using standard subscribed

notification mechanism [RFC8639]. The corresponding telemetry data object capability model is created based on ietf-data-object-tags module defined in this document.

Figure 8 illustrates the advertisment of the list of available target objects using the YANG instance file format [RFC9195]:


```
================ NOTE: '\' line wrapping per RFC 8792 ================

<?xml version="1.0" encoding="UTF-8"?>
<instance-data-set xmlns=\
"urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <name>acme-router-notification-capabilities</name>
  <content-schema>
    <module>ietf-system-capabilities@2020-03-23</module>
    <module>ietf-notification-capabilities@2020-03-23</module>
    <module>ietf-data-export-capabilities@2020-03-23</module>
  </content-schema>
  <!-- revision date, contact, etc. -->
  <description>Defines the notification capabilities of an
    acme-router.The router only has running, and operational
```

```
      datastores. Every change can be reported on-change from
      running, but only config=true nodes and some config=false data
      from operational. Statistics are not reported based on timer
      based trigger and counter threshold based trigger.
    </description>
    <content-data>
      <system-capabilities \
        xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities" \
  xmlns:inc=\
  "urn:ietf:params:xml:ns:yang:ietf-notification-capabilities" \
  xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
        <datastore-capabilities>
          <datastore>ds:operational</datastore>
          <per-node-capabilities>
            <node-selector>\
  /if:interfaces/if:interface/if:statistics/if:in-errors\
  </node-selector>
            <sec:self-describing-capabilities>
              <sec:opm-tag>metric</sec:opm-tag>
              <sec:metric-type>loss</sec:metric-type>
            </sec:self-describing-capabilities>
          </per-node-capabilities>
        </datastore-capabilities>
      </system-capabilities>
    </content-data>
  </instance-data-set>
```

Figure 8: List of Available Target Objects

With telemetry data tagging information carried in the telemetry data tagging Advertisement, the subscriber identifies targeted data object and associated data path to the datastore node and sends a standard establish-subscription RPC [RFC8639] to subscribe tagged data objects

that are interests to the client application from the publisher. Alternatively, the subscriber can query data object tag list from somewhere (e.g., the network device, or offline document) using ietf-data-object-tags module defined in this document and fetch tagged data objects and associated data path to the datastore node and sends a standard establish-subscription RPC [RFC8639] to subscribe tagged data objects that are interests to the client application from the

publisher.

```
================ NOTE: '\' line wrapping per RFC 8792 ================

 <netconf:rpc message-id="101"
     xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
   <establish-subscription
       xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifica\
tions"
       xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
     <yp:datastore
         xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
       ds:operational
     </yp:datastore>
     <yp:datastore-xpath-filter
        xmlns:ex="https://example.com/sample-data/1.0">
       /if:interfaces/if:interface/if:statistics/if:in-errors
     </yp:datastore-xpath-filter>
     <yp:periodic>
       <yp:period>500</yp:period>
     </yp:periodic>
   </establish-subscription>
 </netconf:rpc>
```

The publisher returns specific object types of operational state (e.g., in-errors statistics data) subscribed by the client.

Appendix D. Changes between Revisions

Editorial Note (To be removed by RFC Editor)

v05 - v06

* Additional Editorial changes;

* Use the folding defined in [RFC8792].

v04 - v05

* Add user tag formating clarification;

* Provide guidance to the Designated Expert for evaluation of YANG Data Object Tag registry and YANG Data Object Tag prefix registry.

* Update the figure 1 and figure 2 with additional tags.

* Security section enhancement for user tag managment.

* Change data object name into name in the module.

* Other Editorial changes to address Adrian's comments and comments during YANG docotor review.

* Open issue: Are there any risks associated with an attacker adding or removing tags so that a requester gets the wrong data?

v03 - v04

* Remove histogram metric type tag from metric type tags.

* Clarify the object tag and property tag,metric tag are mutual exlusive.

* Clarify to have two optional node tags (i.e.,object tag and property tag) to indicate relationship between data objects.

* Update targeted data object collection example.

v02 - v03

* Additional Editorial changes.

* Security section enhancement.

* Nits fixed.

v01 - v02

* Clarify the relation between data object, object tag, property tag and metric tag in figure 1 and figure 2 and related description;

* Change Metric Group into Metric Type in the YANG model;

* Add 5 metric types in section 7.2;

v00 - v01

* Merge self-describing data object tag use case section into introduction section as a subsection;

* Add one glossary section;

* Clarify the relation between data object, object tag, property tag and metric tag in Self-Describing Data Object Tags Use Case section;

* Add update to RFC8407 in the front page.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: bill.wu@huawei.com

Benoit Claise
Huawei
De Kleetlaan 6a b1
1831 Diegem
Belgium
Email: benoit.claise@huawei.com

Peng Liu
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing
Email: liupengyjy@chinamobile.com

Zongpeng Du
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing
Email: duzongpeng@chinamobile.com

Mohamed Boucadair
Orange

35000 Rennes
France
Email: mohamed.boucadair@orange.com